**Virtual paint using opencv**

**Abstract:**

In this project, the Python MediaPipe module will be used to create a real-time hand gesture recognition system. Using hand motions recorded by the webcam, the system enables the user to create various objects including lines, rectangles, and circles as well as erase them. A GUI with a number of tools that the user may choose from using hand gestures is also included in the system. When choosing a tool, the MediaPipe Hand Tracking module tracks the user's fingertip and measures its location in relation to the GUI's tools. The "draw" tool allows users to sketch by holding up their index finger; the system detects the gesture by determining if the index finger is elevated over a predetermined level. The system also allows the user to change the thickness of the shapes and the erase tool by modifying a variable in the code. Overall, this project provides a simple yet effective way to interact with a digital drawing board using hand gestures.

**Chapter 1**

**Introduction**

Our popular trending app makes it possible to sketch on air. open-source software, namely Python. As an open source computer vision and machine learning programme, software for learning. The library contains more than 2400 of the top algorithms, including a complete collection of traditional and cutting-edge computer vision and machine learning methods. The majority of these algorithms are used to extract 3D ones, monitor camera motions, identify objects, categorise human actions in films, and detect and recognise faces.

One of the high-level, all-purpose programming languages is Python. The major goal of the object-oriented approach is to assist programmers in producing clean, comprehensible code for both small- and large-scale projects. The morphological procedures that we are carrying out in this project are a series of procedures that use shapes to process photos. These transform an input image into an output image by adding a structural feature. The two most fundamental morphological processes are erosion and dilation.

Virtual Paint using OpenCV is a fascinating project that leverages computer vision and image processing techniques to simulate the act of painting in a digital environment. OpenCV (Open Source Computer Vision) is an open-source library that is widely used for computer vision tasks such as object detection, face recognition, and image processing. In this project, OpenCV is used to track the motion of an object and to paint in real-time on a virtual canvas.

The project involves using a camera to capture a live video feed and then using image processing algorithms to identify the object's location. The user can select the color of their choice and use the object to draw on the canvas. The program detects the object's motion and draws a line on the canvas, giving the impression that the user is actually painting.

This project can be used as a fun and interactive tool for learning, teaching, and entertainment purposes. It can be used as an educational tool for teaching children about colors and shapes or as a creative tool for artists to create digital paintings. Additionally, this project can be used to create interactive installations for events, museums, and exhibitions.

The project's potential uses are endless, and the only limit is one's imagination. In this project, we will explore the development of virtual paint using OpenCV, providing a step-by-step guide for anyone interested in implementing it.

# **CHAPTER 2:**

**Literature Survey:**

The Real time tracking using finger and contour detection for gesture recognition using OpenCV:Gestures are important for communicating information among the human. Nowadays new technologies of Human Computer Interaction (HCI) are being developed to deliver user's command to the robots. Users can interact with machines through hand, head, facial expressions, voice and touch. The objective of this paper is to use one of the important modes of interaction i.e. hand gestures to control the robot or for offices and household applications. Hand gesture detection algorithms are based on various machine learning methods such as neural networks, support vector machine, and Adaptive Boosting (AdaBoost). Among these methods, AdaBoost based hand-pose detectors are trained with a reduced Haar-like feature set to make the detector robust.

“Object Detection Using Image Processing:a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates.Object Recognition is a technique used in the field of computer. It is assumed to be one of the most difficult and challenging tasks in computer. Many methods have been proposed in the past, and a model with new techniques which is not only fast but also reliable. Easynet model has been compared with many other models as well.At the time of prediction, our model bring forth scores for the presence of the object in a particular category.

An Introduction to Computer Vision and Real Time Deep Learning – based Object Detection:The main focus of object detection, one of the most challenging problems in computer vision (CV), is to predict a set of bounding boxes and category labels for each object of interest in an image or in a point cloud. As such, object detection has a variety of exciting downstream applications such as self-driving cars, checkout-less shopping, smart cities, cancer detection, and more. This field has been revolutionized by deep learning over the past five years, where during this time, two-stage approaches to object detection have given way to simpler, more efficient, one-stage models. Mean average precision (mAP) on benchmark problems such as the COCO Object Detection dataset has improved almost 4X over the course of five years from 15% (Fast RCNN, a two-stage approach) to 55% (EfficientDet7x, a one-stage approach).

# **Chapter 3:**

**3.1.Methodology:**

The frames are read and convert the captured frames to HSV color space (Easy for color detection). • Make the canvas frame and put the respective link buttons on it. • Now, Set the track bar values for finding the mask of the colored marker. • Preprocessing of the mask is done with morphological operations (Eroding & Dilation). • The next step goes on like this by, Detecting the contours, finding the center coordinates of large contour and keep storing them in the array for next frames (Arrays for drawing points on the canvass). • Finally, draw the points stored in an array on the frames and canvas.

# **Chapter 4:**

# **PROGRAM IMPLEMENTATION:**

**Sample code:**

# import mediapipe as mp

# import cv2

# import numpy as np

# import time

# #contants

# ml = 150

# max\_x, max\_y = 250+ml, 50

# curr\_tool = "select tool"

# time\_init = True

# rad = 40

# var\_inits = False

# thick = 4

# prevx, prevy = 0,0

# #get tools function

# def getTool(x):

# if x < 50 + ml:

# return "line"

# elif x<100 + ml:

# return "rectangle"

# elif x < 150 + ml:

# return"draw"

# elif x<200 + ml:

# return "circle"

# else:

# return "erase"

# def index\_raised(yi, y9):

# if (y9 - yi) > 40:

# return True

# return False

# hands = mp.solutions.hands

# hand\_landmark = hands.Hands(min\_detection\_confidence=0.6, min\_tracking\_confidence=0.6, max\_num\_hands=1)

# draw = mp.solutions.drawing\_utils

# # drawing tools

# tools = cv2.imread("tools.png")

# tools = tools.astype('uint8')

# mask = np.ones((480, 640))\*255

# mask = mask.astype('uint8')

# '''

# tools = np.zeros((max\_y+5, max\_x+5, 3), dtype="uint8")

# cv2.rectangle(tools, (0,0), (max\_x, max\_y), (0,0,255), 2)

# cv2.line(tools, (50,0), (50,50), (0,0,255), 2)

# cv2.line(tools, (100,0), (100,50), (0,0,255), 2)

# cv2.line(tools, (150,0), (150,50), (0,0,255), 2)

# cv2.line(tools, (200,0), (200,50), (0,0,255), 2)

# '''

# cap = cv2.VideoCapture(0)

# while True:

# \_, frm = cap.read()

# frm = cv2.flip(frm, 1)

# rgb = cv2.cvtColor(frm, cv2.COLOR\_BGR2RGB)

# op = hand\_landmark.process(rgb)

# if op.multi\_hand\_landmarks:

# for i in op.multi\_hand\_landmarks:

# draw.draw\_landmarks(frm, i, hands.HAND\_CONNECTIONS)

# x, y = int(i.landmark[8].x\*640), int(i.landmark[8].y\*480)

# if x < max\_x and y < max\_y and x > ml:

# if time\_init:

# ctime = time.time()

# time\_init = False

# ptime = time.time()

# cv2.circle(frm, (x, y), rad, (0,255,255), 2)

# rad -= 1

# if (ptime - ctime) > 0.8:

# curr\_tool = getTool(x)

# print("your current tool set to : ", curr\_tool)

# time\_init = True

# rad = 40

# else:

# time\_init = True

# rad = 40

# if curr\_tool == "draw":

# xi, yi = int(i.landmark[12].x\*640), int(i.landmark[12].y\*480)

# y9 = int(i.landmark[9].y\*480)

# if index\_raised(yi, y9):

# cv2.line(mask, (prevx, prevy), (x, y), 0, thick)

# prevx, prevy = x, y

# else:

# prevx = x

# prevy = y

# elif curr\_tool == "line":

# xi, yi = int(i.landmark[12].x\*640), int(i.landmark[12].y\*480)

# y9 = int(i.landmark[9].y\*480)

# if index\_raised(yi, y9):

# if not(var\_inits):

# xii, yii = x, y

# var\_inits = True

# cv2.line(frm, (xii, yii), (x, y), (50,152,255), thick)

# else:

# if var\_inits:

# cv2.line(mask, (xii, yii), (x, y), 0, thick)

# var\_inits = False

# elif curr\_tool == "rectangle":

# xi, yi = int(i.landmark[12].x\*640), int(i.landmark[12].y\*480)

# y9 = int(i.landmark[9].y\*480)

# if index\_raised(yi, y9):

# if not(var\_inits):

# xii, yii = x, y

# var\_inits = True

# cv2.rectangle(frm, (xii, yii), (x, y), (0,255,255), thick)

# else:

# if var\_inits:

# cv2.rectangle(mask, (xii, yii), (x, y), 0, thick)

# var\_inits = False

# elif curr\_tool == "circle":

# xi, yi = int(i.landmark[12].x\*640), int(i.landmark[12].y\*480)

# y9 = int(i.landmark[9].y\*480)

# if index\_raised(yi, y9):

# if not(var\_inits):

# xii, yii = x, y

# var\_inits = True

# cv2.circle(frm, (xii, yii), int(((xii-x)\*\*2 + (yii-y)\*\*2)\*\*0.5), (255,255,0), thick)

# else:

# if var\_inits:

# cv2.circle(mask, (xii, yii), int(((xii-x)\*\*2 + (yii-y)\*\*2)\*\*0.5), (0,255,0), thick)

# var\_inits = False

# elif curr\_tool == "erase":

# xi, yi = int(i.landmark[12].x\*640), int(i.landmark[12].y\*480)

# y9 = int(i.landmark[9].y\*480)

# if index\_raised(yi, y9):

# cv2.circle(frm, (x, y), 30, (0,0,0), -1)

# cv2.circle(mask, (x, y), 30, 255, -1)

# op = cv2.bitwise\_and(frm, frm, mask=mask)

# frm[:, :, 1] = op[:, :, 1]

# frm[:, :, 2] = op[:, :, 2]

# frm[:max\_y, ml:max\_x] = cv2.addWeighted(tools, 0.7, frm[:max\_y, ml:max\_x], 0.3, 0)

# cv2.putText(frm, curr\_tool, (270+ml,30), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,0,255), 2)

# cv2.imshow("paint app", frm)

# if cv2.waitKey(1) == 27:

# cv2.destroyAllWindows()

# cap.release()

# break

# 

# **Output:**

# **CHAPTER 5**

**5.1.Testing**

Camera calibration: The first step is to calibrate the camera to ensure that it is properly configured. This can be done using a chessboard pattern and the OpenCV camera calibration function.

Object detection: Once the camera is calibrated, the system needs to be able to detect the object being used for painting. This can be done using color segmentation and contour detection techniques.

Paint simulation: The system should be able to simulate the painting process based on the detected object and the user's movements. This involves tracking the object's position and orientation and using it to create strokes on a virtual canvas.

User interaction: The system should allow the user to interact with the virtual canvas and adjust the painting parameters such as color, brush size, and stroke opacity.

Performance testing: Finally, the system's performance should be tested to ensure that it is responsive and accurate. This can be done by testing the system with different types of objects, painting styles, and user movements.

# **CHAPTER 6**

## **SOFTWARE ENVIRONMENT**

### **6.1 Python**

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generallyare smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

 Machine Learning

 GUI Applications (like Kivy, Tkinter, PyQt etc. )

 Web frameworks like Django (used by YouTube, Instagram, Dropbox)

 Image processing (like Opencv, Pillow)

 Web scraping (like Scrapy, BeautifulSoup, Selenium)

 Test frameworks

 Multimedia

### **6.2 Advantages of Python**

Let’s see how Python dominates over other languages.

#### **1. Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-

generation,unit-testing, webbrowsers, threading, databases,CGI, email, image manipulation, andmore. So, we don’t have to write the complete code for that manually.

#### **2. Extensible**

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### **3. Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

#### **4. Improved Productivity**

The language’s simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### **5. IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### **6. Simple and Easy**

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### **7. Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory**.** This further aids the readability of the code.

#### **8. Object-Oriented**

This language supports both the procedural and object**-** oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

#### **9. Free and Open-Source**

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### **10. Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called WriteOnce Run Anywhere (WORA)**.** However, you need to be careful enough not to include any system-dependent features.

#### **11. Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

### **6.3 Advantages of Python Over Other Languages**

#### **1. Less Coding**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### **2. Affordable**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

#### **3. Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machinelearning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### **6.4 Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### **1. Speed Limitations**

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### **2. Weak in Mobile Computing and Browsers**

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### **3. Design Restrictions**

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck- typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

#### **4. Underdeveloped Database Access Layers**

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### **5. Simple**

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

### **6.5 History of Python**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde&Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voorWiskundeen Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or

begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

### **(a) Learn Linear Algebra and Multivariate Calculus**

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### **(b) Learn Statistics**

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that youneedto learn it!!!Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also,

Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

### **(c) Learn Python**

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as Fork Python available Free on GeeksforGeeks.

#### **Python Development Steps**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others.

It wasalso object orientedandhad a module system.Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python

3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the

13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

· Print is now a function

· Views and Iterators instead of lists

· The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.

· There is only one integer type left, i.e. int. long is int as well.

· The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.

· Text Vs. Data Instead Of Unicode Vs. 8-bit

#### **Purpose**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

#### **Python**

Python is an interpreted high-level programming language for general- purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

· **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

· **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code.

Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

### **6.11 Modules Used in Project**

#### **Numpy**

Numpy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

 A powerful N-dimensional array object

 Sophisticated (broadcasting) functions

 Tools for integrating C/C++ and Fortran code

 Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using

Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Opencv:**

OpenCV is a popular open-source computer vision and image processing library that provides a wide range of image processing algorithms and techniques. It is written in C++ and supports multiple programming languages such as Python, Java, and C#. OpenCV can be used for a variety of applications such as face detection, object recognition, image segmentation, and more.

Here are some key concepts and theories used in OpenCV:

Image processing: OpenCV provides a variety of image processing techniques such as filtering, thresholding, morphological operations, and edge detection. These techniques can be used to enhance images and extract useful information.

Computer vision: OpenCV supports a wide range of computer vision algorithms such as feature detection, object tracking, stereo vision, and camera calibration. These algorithms can be used to extract information from images and video streams.

Machine learning: OpenCV includes several machine learning algorithms such as support vector machines (SVM), k-nearest neighbors (k-NN), and neural networks. These algorithms can be used for tasks such as object recognition and classification.

Feature extraction: OpenCV includes several feature extraction algorithms such as SIFT, SURF, and ORB. These algorithms can be used to detect and extract distinctive features from images for use in tasks such as object recognition.

Image segmentation: OpenCV provides several techniques for segmenting images into regions, such as thresholding, watershed segmentation, and mean-shift segmentation. These techniques can be used for tasks such as object detection and tracking.

Deep learning: OpenCV includes support for deep learning frameworks such as TensorFlow and PyTorch. This enables users to build and train deep learning models for a variety of tasks such as object detection and recognition.

In summary, OpenCV provides a comprehensive set of tools and techniques for image processing, computer vision, and machine learning. Its flexibility and ease of use make it a popular choice for developers and researchers in a wide range of industries.

#### Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

### How to Install Python on Windows and MAC

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system**.** So the steps below are to install python version

3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

### 6.12.1 Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https:/[/w](http://www.python.org/)w[w.python.org](http://www.python.org/)



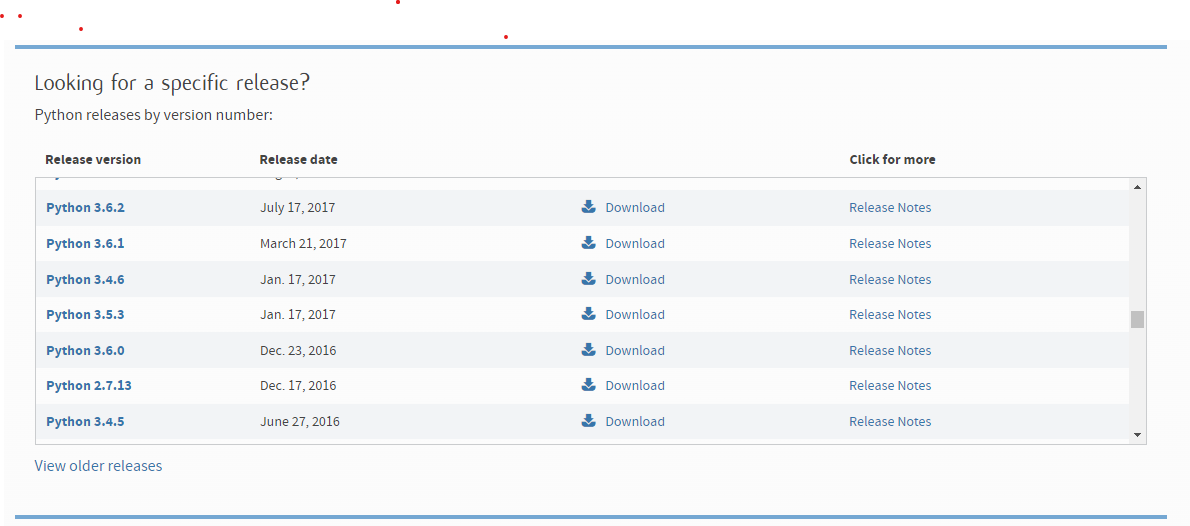
**Fig 6.1**. Search for Windows to Python.Org



**Fig 6.2.** Download the python

**Step 2:** Click on the Download Tab.

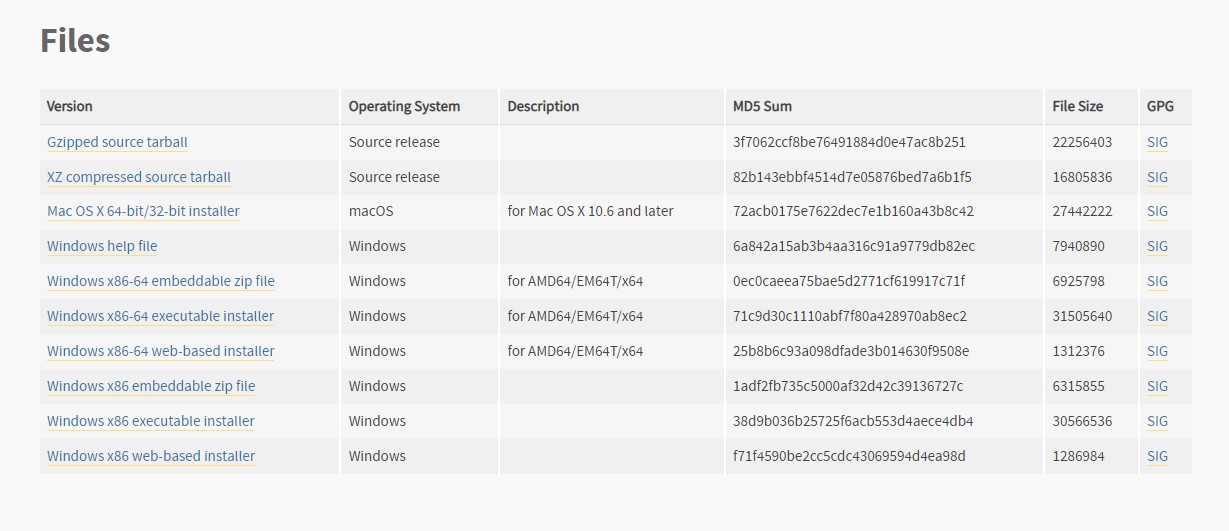
**Step 3:** You can either select the Download Python for windows 3.6.0 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.6.0



**Fig 6.3**. Different versions of python

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



**Fig 6.4.** Different types of python files

* + - To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

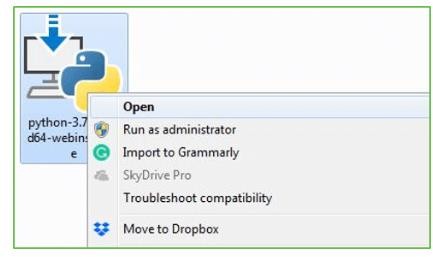
•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Fig 6.5**. Open the python file

**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7to PATH.



**Fig 6.6**. Installation of python

**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



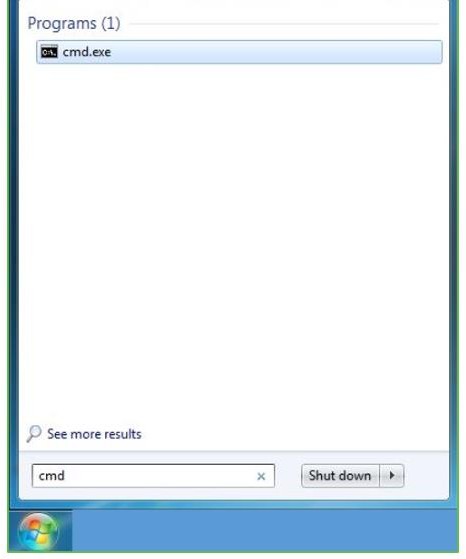
**Fig 6.7**. Successful Installation of python

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes. Verify the Python Installation

**Step 1:** Click on Start

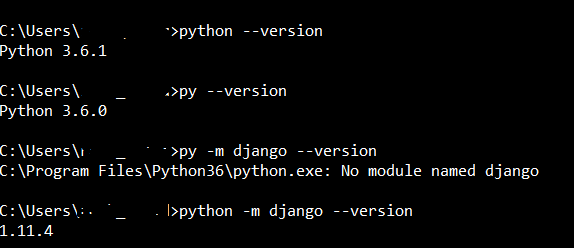
**Step 2:** In the Windows Run Command, type “cmd”.



**Fig 6.8**. Open The Command prompt

**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python – V** and press Enter.



**Fig 6.9.** Output of the python

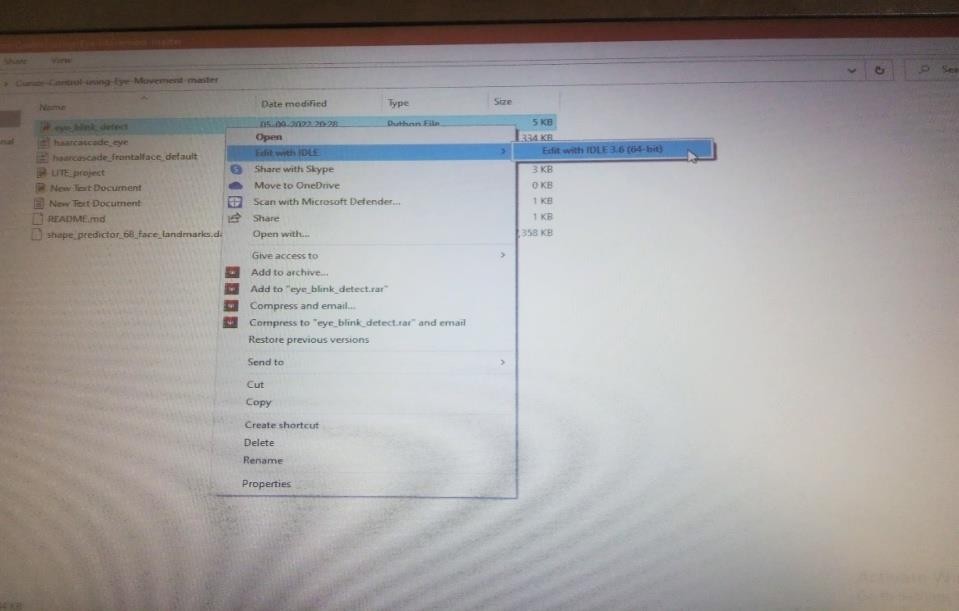
**Step 5:** You will get the answer as 3.7.0

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

**Step 1:** Click on Start

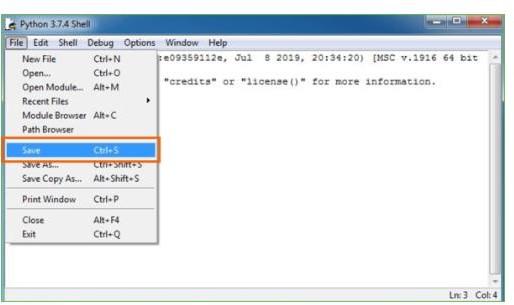
**Step 2:** In the Windows Run command, type “python idle”.



**Fig 6.10.** Edit The IDLE 3.7

**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Fig 6.11.** Save the python files

**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I havenamed the files as Hey World.

**Step 6:** Now for e.g. enter print

# **CHAPTER 7**

## **CONCLUSION AND FUTURE SCOPE**

### **7.1.Conclusion**

In conclusion, the project successfully used the Mediapipe and OpenCV libraries to construct a real-time hand-tracking sketching application. By holding their palm over the matching tool icons shown on the screen, users of the programme may pick drawing tools. Using their index finger, the user may create freehand drawings, lines, rectangles, and circles. An rubber tool is also provided in the programme to clean the sketching canvas. With further features and optimisations, the project—which offers a fun and dynamic method to sketch with hand gestures—could be even better.

### **7.2.Future Scope:**

Integration with other applications: The project could be integrated with other applications, such as email clients or calendar applications, to provide users with a seamless experience. This could involve developing APIs or other integration mechanisms.

Improved accuracy: While the project has achieved high accuracy in detecting emotions from text, there is always room for improvement. Future work could focus on refining the algorithms used to analyze text and improving the accuracy of emotion detection.

Expansion to other languages: Currently, the project supports emotion detection in English text only. Future work could focus on expanding the project to other languages, which would require developing language-specific models for emotion detection.

Real-time emotion detection: The current project analyzes emotion in text that has already been written. Future work could focus on developing a real-time emotion detection system that can analyze emotions in spoken language or in real-time text-based conversations.

Use of machine learning: The project could benefit from the use of machine learning techniques, such as deep learning or reinforcement learning, to improve the accuracy of emotion detection and to enable more sophisticated analysis of emotions in text.

## **REFERENCES**

1. S Guennouni, A Ahaitouf and A Mansouriss , “Multiple object detection using Open CV on an embedded platform”, 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), 2014, pp. 374-377.
2. G. Chandan, Mohana A.H Jain “The Real Time Object Detection and Tracking Using Deep Learning and OpenCV”, 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 2018, pp. 1305-1308.
3. Druzhkov, P.N., Erukhimov, V.L., Zolotykh, N.Y “New object detection features in the OpenCV library”, Pattern Recognit. Image Anal. 21, 384(2011).
4. R. M. Guravv , “The Real time tracking using finger and contour detection for gesture recognition using OpenCV”, 2015 International Conference on Industrial Instrumentation and Control (ICIC), 2015, pp. 974-977.
5. Culjak, D. Abram and T. Pribanic, H. Dzapo and Cifrek, “A brief introduction to OpenCV”, 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 1725-1730.
6. Fares Jalled and Ilia Voronkov, “Object Detection Using Image Processing”, 2016 ariv, 23 Nov 2016.
7. Liangg Dai and James Shanahan, “An Introduction to Computer Vision and Real Time Deep Learning – based Object Detection”, ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 2020, Pages 3523 – 3524

8. Rajeshwar Kumar Dewangan and Yamini Chouhan, “A Review on Object Detection using Open CV Method”, 2020 International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 09 | Sep